

REMARKS

The present application was filed on February 8, 2000 with claims 1-25. Claims 1, 13 and 25 are independent claims. In the outstanding Office Action, the Examiner: (i) objected to the present specification; (ii) rejected claims 1-6, 8-10, 12-18 and 20-25 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,356,933 to Mitchell et al. (hereinafter “Mitchell”); (iii) rejected claims 5, 6, 10, 17, 18 and 22 under 35 U.S.C. §103(a) as being unpatentable over Mitchell in view of U.S. Patent No. 5,838,906 to Doyle et al. (hereinafter “Doyle”); and (iv) rejected claims 7, 11, 19 and 23 under 35 U.S.C. §103(a) as being unpatentable over Mitchell in view of Doyle in further view of U.S. Patent No. 6,170,019 to Dresel et al. (hereinafter “Dresel”).

In this response, Applicants respectfully traverse the various §102(e) and §103(a) rejections of claims 1-25.

Regarding the objection to the present specification, Applicants have amended same to provide further clarification. Accordingly, withdrawal of the objection is respectfully requested.

Regarding the §102(e) rejection of claims 1-6, 8-10, 12-18 and 20-25, the Office Action contends that Mitchell discloses all of the claim limitations recited in the subject claims. Applicants respectfully assert that Mitchell fails to teach or suggest all of the limitations in claims 1-6, 8-10, 12-18 and 20-25, for at least the reasons presented below.

It is well-established law that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). Applicants assert that the rejection based on Doyle does not meet this basic legal requirement, as will be explained below.

The claimed invention, as recited for example in independent claim 1, provides a method for use in a client/server system of reducing interactions between a client and server in association with an application being accessed by the client at the server. The method comprises the steps of: configuring the server to store a model associated with the application and to execute view-generating and controller logic associated with the application; and configuring the client to store at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application, wherein one or more portions

of the application are performed at the client without the client having to interact with the server. Independent claims 13 and 25 recite similar limitations.

As explained at page 3, line 22-27, of the present specification: “[t]he invention addresses performance by employing a dual-MVC approach, in which a subset of the application’s Model-View-Controller reside on the client, and the full Model-View-Controller and View-Generating-Logic reside on the server, thereby reducing the number of required server interactions.” FIG. 3 of the present application illustrates such an inventive dual-MVC approach.

Thus, the claimed invention recites that the server “store[s] a model associated with the application and execute[s] view-generating and controller logic associated with the application,” and the client “store[s] at least a subset of the model associated with the application and execute[s] at least a subset of the view-generating and controller logic associated with the application.”

While Mitchell discloses a system for transmitting interactive application data between a client and a server using markup language, Mitchell does not disclose the dual-MVC approach of the claimed invention. That is, among other deficiencies, Mitchell does not disclose having client “store at least a subset of the model associated with the application,” as in the claimed invention.

The Office Action relies on Mitchell at column 1, lines 5-10, which states:

The present invention relates to client-server networks and, in particular, to methods and apparatus for remotely executing an application and displaying application output.

and at column 2, lines 27-33, which states:

Thus, the present invention overcomes many of the problems faced by traditional approaches outlined above. User interface, event handling and screen rendering logic stay on the client, thus dramatically reducing network traffic and latency. The entire user interface and how that interface connects to application components on the server are sent as a pure data description to the client (rather than code).

Independent claims 1, 13 and 25 clearly state that a client “store[s] at least a subset of the model associated with the application.” Mitchell does not maintain an application model on the client. Mitchell discloses that “user interface, event handling and screen rendering logic stay on the client,” but not an application model.

The Office Action also cites Mitchell at column 2, lines 11-50:

The present invention provides a mechanism by which the user interface portion of the application can be delivered to the computer user either on the same machine on which the application is executing or on another machine remote from the machine executing the application. The invention separates the user interface from the underlying application enabling the user interactive portion of the application to be extremely simple. The invention also permits the user interactive portion to be deployed on a wide range of client hardware environments without bringing with it all the required logic for performing the functionality of a particular application. These features give the user the effect of directly interacting with whole application even though the main part of the application is potentially running somewhere else.

Thus, the present invention overcomes many of the problems faced by traditional approaches outlined above. User interface, event handling and screen rendering logic stay on the client, thus dramatically reducing network traffic and latency. The entire user interface and how that interface connects to application components on the server are sent as a pure data description to the client (rather than code). This description is "interpreted" by the client to render the graphics user interface (GUI) and connect to the application (through the transfer of state) running either in the same process space (same machine) or on the server (remote machine).

Because the server can communicate with a particular application client with simply a data description, no additional code needs to be installed on the client machine. An application-independent client process (AICP) reads the description and presents that description to the user as a typical client user interface. Therefore, the AICP can communicate with an unlimited number of server applications with a new data file description for each program (which can be cached automatically as required or as specified by the client). No application specific administration is required for executing an AICP-deployed application using this approach.

As is made clear, the "pure data description" sent to the client in Mitchell is a description of the GUI. This description is "interpreted" by the client to render the graphics user interface (GUI) and connect to the application (through the transfer of state) running either in the same process space (same machine) or on the server (remote machine). This indicates that the application can run on the client or the server, not both. However, there is no discussion of application model on the client.

The Office Action also cites Mitchell at column 4, line 52, through column 5, line 30:

The transaction processor 430 is a program that receives information from the web server application 132 via a common gateway interface (not shown), interprets the information to determine whether a specific instance of an AISP 134 is required, and launches the instance AISP 134 to further process the request received from the AICP 114.

Referring to FIG. 3, the present invention includes the AICP 114 and the AISP 134.

The AICP 114 renders the graphical user interface (GUI) that is displayed to the user on the user interface 116. The AICP 114 also maintains a relationship between the control objects displayed on the user interface 116 and the application components 136 maintained on the web server 130. The AISP 134 tracks the state of the application components 136 along with the control objects displayed on the user workstation 110 that require updates of these application components. Whenever the state changes on either the client (control state) or the server (component state), the AICP 114 and AISP 134 take appropriate action based on the data description that defines the relationship between the GUI controls and the server application components 136 (hereafter referred to as server components) they represent.

Referring also to FIG. 4, the relationship 446 between the control objects 624 displayed on the user interface 116 of the user workstation 110 and the server components 136 maintained on the web server 130 include data that describes an explicit relationship between their respective object interfaces. This data will hereafter be referred to as a connection. The AICP and AISP contain logic that can interpret connections that relate a visual control to an application component.

For example, a scroll bar control is representative of a type of control object that can be displayed on the user interface 116 of the user workstation 110. The scroll bar control can be associated with the value of an application component, such as the temperature of an industrial process. As the server application detects a temperature change, the state of the Application Components 136 is changed and these state changes 330 are forwarded to the client. The scrollbar is subsequently redrawn by the AICP to reflect the new value. Likewise, if a scroll bar is connected to an Application Component 136 that controls a thermostat, then when the user interacts with the scroll bar on the user interface 116, the state change is transmitted to the Web Server Application Program 132 which would change the state of the appropriate Application Component 136 which would subsequently set the thermostat.

This portion of Mitchell discusses the GUI (including “control objects”). The client state that is tracked is the control (GUI) state. The application components are clearly on the server. There is no discussion of application model objects on the client, only controls (view elements).

Accordingly, Applicants assert that independent claims 1, 13 and 25, as well as the claims which depend therefrom, are patentable over Mitchell and therefore allowable. Such dependent claims also recite patentable subject matter in their own right.

Regarding the §103 rejections to claims 5-7, 10, 11, 17-19, 22 and 23, Applicants respectfully assert that such dependent claims are patentable over the Mitchell/Doyle combination and/or the Mitchell/Doyle/Dresel combination for at least the reasons given above with respect to independent claims 1 and 13. Both Doyle and Dresel fail to remedy the deficiencies of Mitchell. However, Applicants also assert that such dependent claims also recite patentable subject matter in

their own right.

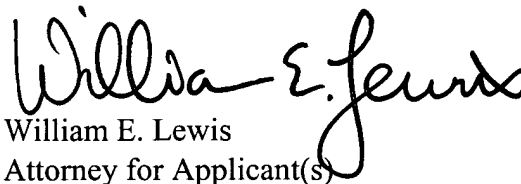
Further, there is a clear lack of motivation to combine Mitchell, Doyle and Dresel. Other than very general and conclusory statements in the Office Action, there is nothing in the three references that reasonably suggests why one would actually combine the teachings of these two references.

The Federal Circuit has stated that when patentability turns on the question of obviousness, the obviousness determination “must be based on objective evidence of record” and that “this precedent has been reinforced in myriad decisions, and cannot be dispensed with.” In re Lee, 277 F.3d 1338, 1343 (Fed. Cir. 2002). Moreover, the Federal Circuit has stated that “conclusory statements” by an examiner fail to adequately address the factual question of motivation, which is material to patentability and cannot be resolved “on subjective belief and unknown authority.” Id. at 1343-1344.

Applicants submit that the statements given by the Examiner in the present Office Action are based on the type of “subjective belief and unknown authority” that the Federal Circuit has indicated provides insufficient support for an obviousness rejection. More specifically, the Examiner fails to identify any objective evidence of record which supports the proposed combination.

In view of the above, Applicants believe that claims 1-25 are in condition for allowance, and respectfully request withdrawal of the §102(e) and §103(a) rejections.

Respectfully submitted,



William E. Lewis
Attorney for Applicant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946

Date: February 24, 2005